Discrete Optimization 9 (2012) 241–248

Contents lists available at SciVerse ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

An asymptotically optimal online algorithm to minimize the total completion time on two multipurpose machines with unit processing times

Dvir Shabtay *, Shlomo Karhi

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 4 March 2012
Received in revised form 10 June 2012
Accepted 27 July 2012
Available online 18 August 2012

Keywords:
Multipurpose machine scheduling
Online scheduling
Total completion time
Competitive ratio

ABSTRACT

In the majority of works on online scheduling on multipurpose machines the objective is to minimize the makespan. We, in contrast, consider the objective of minimizing the total completion time. For this purpose, we analyze an online-list scheduling problem of n jobs with unit processing times on a set of two machines working in parallel. Each job belongs to one of two sets of job types. Jobs belonging to the first set can be processed on either of the two machines while jobs belonging to the second set can only be processed on the second machine. We present an online algorithm with a competitive ratio of $\rho_{LB} + O(\frac{1}{n})$, where ρ_{LB} is a lower bound on the competitive ratio of any online algorithm and is equal to $1 + \left(\frac{-\alpha+\sqrt{4\alpha^3-\alpha^2+2\alpha-1}}{2\alpha^2+1}\right)^2$ where $\alpha = \frac{1}{3} + \frac{1}{6}\left(116 - 6\sqrt{78}\right)^{1/3} + \frac{(58+3\sqrt{78})^{1/3}}{3(2)^{2/3}} \approx 1.918$. This result implies that our online algorithm is asymptotically optimal.

1. Introduction

We study an online scheduling problem with unit processing times on a set of two multipurpose machines where the objective is to minimize the total completion time. Our problem can be formally stated as follows. A set of n jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ is available at time zero to be processed nonpreemptively on a set of two machines $\mathcal{M} = \{M_1, M_2\}$ working in parallel. The jobs are categorized into two job types according to the set of machines that can process each job. Let $\mathcal{M}_l$ be the set of machines that can process jobs of type l for $l = 1, 2$. We consider the case where $\mathcal{M}_1 = \{M_1, M_2\}$ and $\mathcal{M}_2 = \{M_2\}$. Since machine M_2 can process both job types, it is referred to as a flexible machine, while machine M_1 is referred to as a non-flexible machine. All jobs have the same processing time and, without loss of generality, we assume that processing times are restricted to unity; that is $p_j = 1$ for $j = 1, 2, \ldots, n$ where p_j is the processing time of job J_j. We aim to assign the jobs to the machines such that the *total completion time*, $z = \sum_{j=1}^{n} C_j$, will be minimized, where C_j is the completion time of job J_j for $j = 1, 2, \ldots, n$.

Given an assignment of jobs to machines, let x_i be the number of jobs that have been assigned to machine M_i for $i = 1, 2$. Since all jobs have unit processing time, the completion time of the jth job to be processed on some machine M_i is exactly at time j for $j = 1, \ldots, x_i$. Thus, the total completion time is given by

$$z = \sum_{i=1}^{2}\sum_{j=1}^{x_i} j = \frac{x_1(x_1+1)}{2} + \frac{x_2(x_2+1)}{2}. \quad (1)$$

* Corresponding author. Tel.: +972 8 6461389; fax: +972 8 6472958.
E-mail addresses: dvirs@bgu.ac.il (D. Shabtay), shlomok@post.bgu.ac.il (S. Karhi).

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2012.07.005

We assume that the online version of our problem follows the *online-list* paradigm, where the jobs are ordered in a list and are presented to the scheduler one by one. As soon as a job is presented to the scheduler he knows its type. Then, he has to assign the jobs according to an *online algorithm* where each job has to be irreversibly assigned to some machine before the next job is presented. It is commonly assumed in online scheduling that, in addition to the uncertainty about the job parameters, the scheduler does not know the number of jobs in the list.

In this paper we develop an online algorithm for minimizing the total completion time. In order to evaluate the quality of our online algorithm, the *competitive analysis* evaluation technique presented by Sleator and Tarjan [1] is used. Competitive analysis is a type of worst-case analysis in which the performance of an online algorithm is compared to that of an optimal *offline algorithm*. In *offline scheduling* the scheduler has access to the entire instance of the problem prior to making any scheduling decision. Thus, in our case the scheduler knows in advance both the number of jobs in the list and the type of each job. Therefore, he can apply an *offline algorithm* which takes all the data about the jobs into consideration when making any scheduling decision.

The competitive analysis evaluation technique can be described as follows. Let z be a criterion (objective function) that has to be minimized. For an online Algorithm A, let $z^A(\mathcal{I})$ denote the objective value produced by Algorithm A, for instance $\mathcal{I} \in I$, where I is the set of all possible instances. Further, let OPT be an optimal offline algorithm, and let $z^{\text{OPT}}(\mathcal{I})$ be the corresponding minimum objective value for instance $\mathcal{I}$. We say that Algorithm A is *ρ-competitive* if the condition that $z^A(\mathcal{I}) \le \rho z^{\text{OPT}}(\mathcal{I})$ holds for any input instance $\mathcal{I} \in I$. Moreover, the *competitive ratio* of Algorithm A denoted by ρ_A is the infimum of ρ such that A is ρ-competitive. According to Sleator and Tarjan, an online scheduling problem has a lower bound ρ_{LB} if no online algorithm has a competitive ratio smaller than ρ_{LB}. Moreover, an online algorithm is called optimal if its competitive ratio matches the lower bound of the problem.

Different variants of the online-list scheduling problem on a set of multipurpose machines have been discussed in the literature (e.g., [2–10]), all of which consider the objective of minimizing the makespan. To the best of our knowledge, our paper is the first to consider the total completion time criterion in the context of online-list scheduling on multipurpose machines. Although our analysis is restricted to two machines with unit processing times, we believe that in future studies these results can be generalized.

The rest of the paper is organized as follows. In Section 2 we determine the optimal assignment of jobs to machines in an offline system. Moreover, we show that the problem has a lower bound of $\rho_{\text{LB}} = 1 + \left(\frac{-\alpha+\sqrt{4\alpha^3-\alpha^2+2\alpha-1}}{2\alpha^2+1}\right)^2$, where $\alpha = \frac{1}{3} + \frac{1}{6}\left(116 - 6\sqrt{78}\right)^{1/3} + \frac{\left(58+3\sqrt{78}\right)^{1/3}}{3(2)^{2/3}} \approx 1.918$. This implies that no online algorithm has a competitive ratio smaller than $\rho_{\text{LB}} \approx 1.1573$. In Section 3 we present an online algorithm with a competitive ratio of $\rho = \rho_{\text{LB}} + O\left(\frac{1}{n}\right)$. This result implies that our online algorithm is asymptotically optimal. A summary section concludes our paper.

2. A lower bound on the competitive ratio

The following lemma provides the minimal total completion time value and the optimal job assignment strategy for the *offline* version of our problem.

Lemma 1. *Given an instance $\mathcal{I} \in I$, the optimal job assignment to machines $x^* = (x_1^*, x_2^*)$ for an offline problem is*

$$x^* = \begin{cases} \left(\left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil\right) & \text{if } n_1 \ge n_2 \\ (n_1, n_2) & \text{if } n_1 < n_2, \end{cases} \tag{2}$$

and the minimum value of the total completion time is

$$z^{\text{OPT}}(\mathcal{I}) = \begin{cases} \frac{\lfloor \frac{n}{2} \rfloor \left(\lfloor \frac{n}{2} \rfloor + 1\right)}{2} + \frac{\lceil \frac{n}{2} \rceil \left(\lceil \frac{n}{2} \rceil + 1\right)}{2} & \text{if } n_1 \ge n_2 \\ \frac{n_1 (n_1 + 1)}{2} + \frac{n_2 (n_2 + 1)}{2} & \text{if } n_1 < n_2, \end{cases} \tag{3}$$

where for instance $\mathcal{I}$, n_l represents the number of jobs of type l for $l = 1, 2$ in the set of n jobs.

Proof. The offline problem can be solved by minimizing Eq. (1) subject to $x_2 \ge n_2$ and $x_1 + x_2 = n$, where x_i is restricted to be a non-negative integer for $i = 1, 2$. Since $x_1 + x_2 = n$, we can rewrite Eq. (1) as

$$z(x_1) = \frac{x_1(x_1+1)}{2} + \frac{(n - x_1)(n - x_1 + 1)}{2} = (x_1)^2 - nx_1 + \frac{n(n+1)}{2}. \tag{4}$$

The lemma now follows from the fact that $z(x_1)$ is a convex function whose minimum is at the point where either $x_1 = \left\lceil \frac{n}{2} \right\rceil$ or $x_1 = \left\lfloor \frac{n}{2} \right\rfloor$. □